AIAA-90-3174

ANALYZING TIME DELAYS IN A FLIGHT SIMULATION ENVIRONMENT

R. E. McFarland
NASA Ames Research Center
Moffett Field, CA

J. W. Bunnell
Syre, Inc.
NASA Ames Research Center
Moffett Field, CA

AIAA Flight Simulation Technologies Conference and Exhibit
September 17-19, 1990/Dayton, Ohio

ANALYZING TIME DELAYS IN A FLIGHT SIMULATION ENVIRONMENT

R. E. McFarland*
NASA Ames Research Center, Moffett Field, California
and
J. W. Bunnell**
Syre, Inc., Ames Research Center, Moffett Field, California

Abstract

Transport delay in a multirate flight simulation environment is examined. An equivalent systems model is developed that quantifies the contributions of individual components and their sampled-data interactions. Mathematical algorithms used in the discrete implementation are also considered, because they are important elements of a flight simulation system. The equivalent systems model was used to demonstrate the consistency and accuracy of data obtained in the flight simulation facility at Ames Research Center. It showed that effective time delays in simulation models, including delays in scene presentation to the pilot, are considerably less than might be assumed by casual examination of raw data obtained from component-level experiments.

Introduction

A total simulation system is rarely exercised in obtaining time delay measurements. Notably, compensating algorithms are usually disabled, and drive signals used in an investigation often bear little resemblance to pilot inputs. In Ref. 1 (p. 14), for example, a step signal was sent through an analog-to-digital converter (A/D) to produce a discrete change in a computer generated image (CGI display). The difference between the time of the command and time of the response purportedly measured the system delay. Such an experiment has several features which require interpretation, in order to apply the results to the flight simulation environment.

First, during the A/D conversion process, a step input has a uniform distribution within a sample interval. The signal also has an approximately uniform distribution within the CGI computer, where compensating logic (which usually accounts for asynchronous data transmittal between computers) cannot utilize discontinuous data. The resulting compound distribution contributes as much as two computer cycles of delay. This constitutes a major difference from the actual simulation environment, where inputs are generally sampled such that significant activity does not occur between sample intervals—a fundamental hypothesis of real-time simulation. If the hypothesis fails, then the simulation fails.

Second, in Ref. 1 specifically, since the signal was not integrated, it could not be predicted to the end of the cycle, a function normally handled by an integration process, as will be discussed. This introduced an entire cycle of delay.

Third, recently developed CGI compensation logic, used in flight simulation at Ames Research Center, requires that CGI commands also be representative samples of continuous signals. Although the compensation scheme was developed subsequent to the experiment of Ref. 1, its availability at that time would not have influenced results, because of the continuity requirement. Hence, a repeat of this particular experiment today would still manifest the CGI pipeline delay. However, because of the compensation algorithm, this delay is not observed in flight simulation (over the frequency range pertinent to handling qualities research).

Thus, the data of Ref. 1, although accurate, are nonetheless misleading because they indicate a delay on the order of 150 msec, whereas in the flight simulation environment at Ames the effective delay is more like 15 msec.

A model of flight simulation systems is therefore needed that quantifies the contributions of various delay components and contains the proper algorithmic relationships. For this reason an equivalent systems model (ESM) has been developed. It reconstructs complete data paths

*Aerospace Engineer.

**Simulation Programs Manager. Member AIAA.

in the simulation environment, where both algorithms and sampled-data effects are important elements. To establish a foundation for this material, the sampled-data phenomenon is examined.

From the perspective of a discrete, multirate, man-in-the-loop flight simulation, the sampled-data system discussed in this paper is outlined in Fig. 1. This figure illustrates the sampled-data elements of a simulation that are pertinent to the creation of a discrete realization from a continuous system.

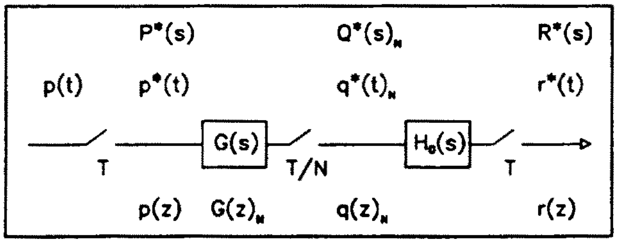

Fig. 1. Sampled-data system.

$G(s)$ represents the multirate simulation model, sampled N times for each input-output (I/O) cycle time T. Because this is a sampled system, the function $G(s)$ consists of the product of a model transfer function $F(s)$ and a data-hold function $H_i(s)$, as required in a data reconstruction process. This will be discussed further in the section "The Discrete Multirate Model." $G(s)$ may, of course, consist of a network of transfer functions, with some resultant equivalent data-hold function. The symbol $p(t)$ represents a pilot input, and $q^*(t)_N$ is a model output at the multirate level of computation. The final output $r^*(t)$ is sampled at the same rate as the pilot input (the I/O rate). This is accomplished in the ESM via a zero-order hold (low-pass filter) shown as $H_0(s)$ in Fig. 1.

Using the above outline, an ESM is developed for the analysis of delays in flight simulation. Models of multirate computer processes using z-transform notation are treated first, using essential elements from sampled-data theory.

Input Sampler

Pilot control inputs are represented in Fig. 1 by the time function $p(t)$, here assumed continuous with Laplace transform $P(s)$. Neglecting, for the moment, converter dynamics and possible prefiltering, the A/D operation consists of a switch that samples the input every T seconds and delivers a pulse train of data to the digital computer. Since the transfer function of a linear system is the Laplace transform of the impulsive response of the system, at the sample points the pulse transfer function is defined by the ordinary Laplace transform.

"The pulse transfer function relates only the output pulse sequence to the input pulse sequence. It does not relate the continuous output to the pulse sequence at the input" (Ref. 2, p. 69). As an example of the sampling process, consider the input sine function,

$$p(t) = \sin(at)$$

Its pulse transfer function is obtained from any table of Laplace transforms,

$$P(s) = a/(s^2 + a^2)$$

and is defined as producing a series of impulses represented by the z-transform

$$p(z) = \mathcal{Z}\{P(s)\} = \frac{\sin(aT)}{z - 2\cos(aT) + z^{-1}}$$

"When a unit-impulse function is applied to a linear system with transfer function $P(s)$, the output of the system is simply the impulse response of the system" (Ref. 3, p. 74). To examine the equivalent pulse train within a digital computer, the z-transform is reduced to a difference equation using a difference operator for powers of z,

$$y_{k+1} = 2\cos(aT)y_k - y_{k-1} + \sin(aT)p_k$$

With quiescent initial conditions and a unit impulse for an input, a trivial induction shows that at the sample points, for all $k \geq 0$,

$$y_k = \sin(akT)$$

Thus, in the process of sampling continuous data, the sampler (perfect A/D converter) does not produce signal delay.

Interpolation

Consider the multirate system shown in Fig. 1, in which N is assumed to be an integer. As will be shown in the section "The Discrete Multirate Model," the output of a multirate system may be related to its input using z-transforms.

When the new sampling rate is higher than the original sampling rate the process is called interpolation, because samples of the original physical process are collated from a reduced set of samples. In Fig. 1 the signal $q^*(t)_N$ is sampled at a rate N times faster than the sampling rate at the input. If $P^*(s)$ is defined as the pulse transform of $p(t)$, the Laplace transform of the intermediate result sampled at the expanded rate, $Q^*(s)_N$, is given by (Ref. 2, pp. 221-222; Ref. 3, pp. 82-85, 353-357)

$$Q^*(s)_N = \sum_{n=0}^{N-1} \left[G(s)e^{nTs/N}\right]^* P^*(s)e^{-nTs/N}$$

The z-transform of this expression is given by

$$
\begin{aligned}
q(z)_N &= \sum_{n=0}^{N-1} \mathrm{Z}\left\{G(s)e^{nTs/N}\right\} p(z)z^{-n/N} \\
&= \sum_{n=0}^{N-1}\sum_{k\geq 0} g[(k+n/N)T]e^{-[k+n/N]Ts}p(z) \\
&= \sum_{m\geq 0} g(mT/N)e^{-mTs/N}p(z) \\
&= G(z)_N p(z)
\end{aligned}
$$

where the development includes the substitution of $m/N = k + n/N$ (Ref. 3, p. 357). Hence, $G(z)_N$ can be obtained from the ordinary z-transform $G(z)$ by the replacements

$$G(z)_N = [G(z)]_{\substack{z=z^{1/N} \\ T=T/N}}$$

Thus, the multirate z-transform of the output of a multirate sampled system is obtained from the ordinary z-transform of the input and the multirate z-transform of the model (with its associated data hold). In terms of a simulation model, therefore, the z-transform representation of one subcycle of the model is sufficient for an analysis of a multirate system, providing the decimated output can be obtained from the multirate output.

Because of this relationship, it is natural to define the multirate cycle time as $\tau = T/N$, where τ represents one subcycle of the model. The z-transform operator is given by

$$z = \cos(\omega T) + j\sin(\omega T) = \cos(N\omega\tau) + j\sin(N\omega\tau)$$

Multirate Signals

The signal $q(z)_N$ represents data at the expanded sample rate. Because of its extended Nyquist limit, $f_N = N/2T$, it contains frequency content beyond the Nyquist limit imposed by the final (output) sampler, $f_T = 1/2T$. As will be shown in the next section, the baseband of interest ($f < fT$) is recovered in the ESM via a low-pass filter. This filter may be a data hold.

Without the low-pass filter, the spectrum of $q(z)_N$ would contain not only the baseband frequencies of interest (i.e., $f < f_T = 1/2T$) at the I/O level of computation, but also images of the baseband centered at harmonics of the original sampling frequency. This extended frequency content would occur at all multiples of the original sampling frequency, limited only by the expanded Nyquist frequency, $Nf_T = f_N$, beyond which nothing may be observed from a discrete model.

In a real-time multirate simulation environment, a nonlinear model may also generate frequencies beyond f_T (up to f_N). Indeed, the reason for implementing a multirate model is usually the high-frequency content generated within a model by internal nonlinearities and feedback paths (which require algorithmic approximations in the convolution to a discrete realization). Hence, it is sometimes observed that a smaller effective cycle time (than the I/O cycle time) is required to attain numerical stability.

The consequences of multirate processes, however, deserve some discussion. If a smaller internal cycle time is required, then the discrete model may produce frequency activity beyond f_T. If these frequencies are present in the outputs of multirate components, then upon decimation to the I/O rate, these signals will alias into the I/O bandwidth and will appear at some other frequency location; this will alter, and possibly destroy, the integrity of the model. This phenomenon was reported in Ref. 4, wherein an "aliasing equation" was provided to show that aliased frequencies may adversely influence the pilot/simulator bandwidths, especially in rotorcraft models.

Good reason exists for implementing a multirate model, providing the contaminating frequencies are filtered at the expanded rate prior to decimation to the I/O rate (see Refs. 4 and 5). In that case, upon decimation, only attenuated signals are aliased into the I/O passband. Filtering algorithms require special attention, however, because they must not adversely influence either phase or gain in the pilot-vehicle bandwidth.

The multirate technique without filtering is often inferior to other techniques for attaining stability. Although it may be attained, the original problem is exchanged for other problems, e.g., aliasing. In certain instances there are alternate solutions for attaining stability; these include algorithmic substitutions, purging the model of components that create frequency content beyond the I/O bandwidth, and antialiasing filters in the model.

The spectrum of $q(z)_N$ is thus of interest because activity beyond f_T influences decimated simulation outputs $r(z)$ via the aliasing phenomenon. However, signals that originate in a model beyond the I/O Nyquist frequency cannot be aliased by the ESM. Fortunately, this is of little consequence in the determination of effective time delay, which involves evaluations only at low frequencies.

The Discrete Multirate Model

The behavior of the input data between sample points is assumed by selecting a data hold. The discrete model $G(z)$ is obtained from the data-hold assumption (or a combination of data holds). The data hold serves as a low-pass filter, which is required in extracting the signal spectrum between sample points. A temporal shift may also accompany a data hold. Three holds are of interest here; they may be obtained from the Newton-Gregory formula (Ref. 2, pp. 31-39). They are given by

$$H_0(s) = \frac{1 - e^{-sT}}{s}$$

$$H_1(s) = \left(\frac{1 - e^{-sT}}{s}\right)^2 \left(\frac{1 + sT}{T}\right)$$

$$H_2(s) = \left(\frac{1 - e^{-sT}}{s}\right)^2 \frac{e^{sT}}{T}$$

$H_0(s)$ is the zero-order hold, in which the input is assumed to remain constant between sample points. It tends to advance the output by a half cycle from the most recent input point, although it purports to advance outputs by a full cycle. Its utility is often overstated because it happens to produce perfect answers (advanced one cycle) for step inputs. $H_1(s)$ is the first-order hold, in which the input is linearly extrapolated beyond the most recent input point. It tends to advance the output by a full cycle. $H_2(s)$ is the triangular hold, in which the input is linearly interpolated during the interval (Ref. 2, p. 288). It delivers an output that is concurrent with the most recent input.

The multirate discrete transfer function is obtained from

$$G_i(z)_N = \mathcal{Z}\{H_i(s)F(s)\}_N$$

where the subscript i is used to keep track of the selected data hold. At the multirate level of computation, $G_i(z)_N$ relates an output to an input. It is important to keep in mind that the selection of the zero-order or first-order data hold produces advanced outputs (a time shift).

All three data holds are useful in creating a $G(z)$ that describes models containing networks of transfer functions that include feedback paths. Whenever a time advance (shift) is not required, the use of the triangular hold is highly recommended. The zero-order hold is not recommended unless phase is immaterial (as in the case of random inputs), or unless it is needed for stability in a feedback loop, in which a first-order hold implementation could produce a negative gain margin (unstable discrete realization).

Because $G_i(z)$ is transformed to $G_i(z)_N$ in the ESM, the data hold is effective over the extended Nyquist limit. Hence, $G_i(z)_N$ tends to match the spectrum of $F(s)$ beyond the I/O Nyquist limit. Because of decimation, however, a different spectrum is observed upon output.

Decimation

The spectrum created from the interpolation process beyond the I/O Nyquist limit must be purged from the mathematical representation prior to observation of the decimated output $r(z)$. "Based upon well known sampling theory, in order to lower the sampling rate and to avoid aliasing at this lower rate, it is necessary to filter the signal with a digital low-pass filter which approximates the ideal characteristic" (Ref. 6, pp. 302-303):

$$L(j\omega) = \begin{cases} 1/N & \omega \le \pi/T \\ 0 & \omega > \pi/T \end{cases}$$

A filter that reasonably attains this characteristic is given by

$$L(z) = \frac{z^{-1}}{N}\sum_{n=1}^{N} z^{n/N} = \frac{1 - z^{-1}}{N(1 - z^{-1/N})}$$

a term we call the rate conversion factor. It converts spectra at the expanded rate to spectra at the I/O rate, and is thus an I/O passband filter. The gain of the rate conversion function $L(z)$,

$$|L(z)| = N^{-1}|\sin(\omega T/2)/\sin(\omega T/2N)|$$

is plotted in Fig. 2 for $N = 2, 3, 4$, and 5. The abscissa of Fig. 2 represents the normalized frequency, with a value of unity being equivalent to the multirate Nyquist frequency (π/τ).

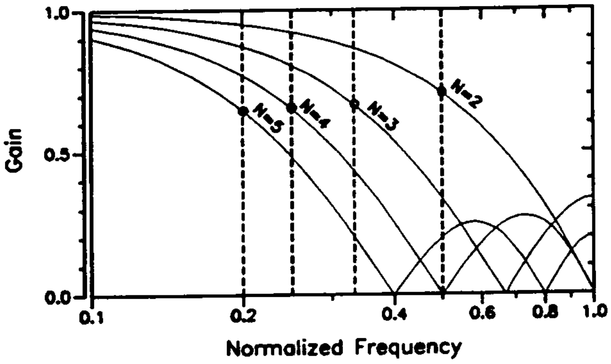

Fig. 2. Conversion gain.

The I/O Nyquist frequency after the decimation process (π/T) is also indicated in Fig. 2; the gain at this frequency approaches $2/\pi$ as N increases. These curves illustrate that the rate conversion factor $L(z)$ does a satisfactory job in attenuating frequency content beyond the I/O Nyquist limit.

In order to relate the decimated output $r(z)$ to the input $p(z)$ we must also include the computer delay per subcycle, $z^{-1/N}$. The multirate portion of the ESM is then represented by the transfer function relationship

$$W(z) = r(z)/p(z) = z^{-1/N}L(z)G_i(z)_N$$

Comparisons

The accuracy of $W(z)$ to predict delay times in typical flight vehicle models can be demonstrated by comparing its results with those obtained using time-series analyses or real-time simulations. To accomplish this, the delays in simulation were established using Monte Carlo techniques, with five random variables. A fifth-order Laplace set of functions was selected as being representative of flight simulation. This set is given by

$$F(s) = \frac{\omega_1^2 \omega_2^2}{(\lambda s + 1)(s^2 + 2\zeta_1 \omega_1 s + \omega_1^2)(s^2 + 2\zeta_2 \omega_2 + \omega_2^2)}$$

The five random variables were given the ranges

$$2\pi < \omega_1 < 4\pi$$

$$4\pi < \omega_2 < 8\pi$$

$$0.1 < \zeta_1,\ \zeta_2 < 0.9$$

$$0.05 < \lambda < 0.5$$

The phase angles, and hence the equivalent time delays, can easily be computed at any given frequency by substituting $j\omega$ for s in the Laplace representations. Selecting the observation frequency as $\omega = \pi(1/2 \text{ Hz})$ produced 300 random time delays, as shown in Figs. 3(a), 3(b), and 3(c) by D_i. Values up to about 600 msec were observed using the selected range of random variables. Time delay is defined in this paper as the negative of the relative phase shift divided by the frequency.

The time-series outputs of the 300 functions were obtained using a cycle time of 30 msec. Multirate simulations were performed, in which the number of multiple loops (N) ranged from one to ten, in sequential simulations. The driving function, an input to the simulations at the I/O rate of 30 msec, was a 1/2-Hz sine wave (permitting comparison with the Laplace computations). The multirate portion of the simulation model was handled by an internal DO LOOP, within which the cycle time was $\tau = T/N$. This was computed for each of the data holds i discussed in this paper, by use of software developed from the techniques of Ref. 7.

The phase angle from each simulation was computed by trigonometric least-squares techniques, in which the phase angle (and resultant time delay) is identified. By differencing the observed time delay from that obtained from the Laplace functions, the relative time delay (produced from the discrete simulation) was computed. The results of these experiments are presented in Figs. 3(d), 3(e), and 3(f). This error is presented as ΔD_i for the various holds.

The first 10 runs of Fig. 3 used $N = 1$, the second 10 runs used $N = 2$, etc. For example, runs 91 through 100 used 10 multiple loops.

The results of this analysis show dependence only on the specific data hold (i) and on the number of multiple loops (N). The results are independent of the functionality (simulation model!) providing only that the equivalent phase shift caused by the discrete realization process (data hold) is isolated. From these data, and other data obtained at different cycle times, the delay produced by multirate simulations (with a reasonable bandwidth) is found to be, to negligible measurement error,

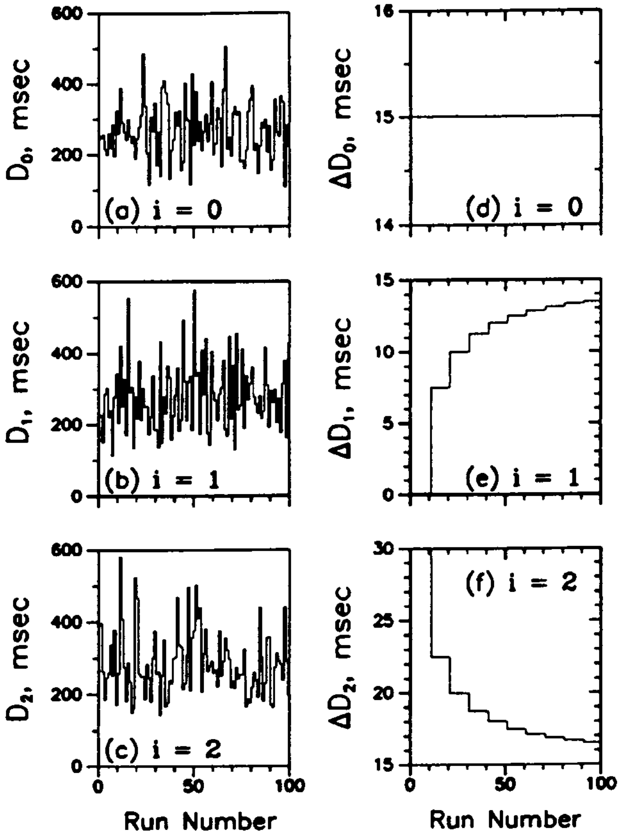

Fig. 3. Monte Carlo results.

$$\Delta D_0 = \frac{1}{2}T$$

$$\Delta D_1 = \frac{1}{2}T(1 - 1/N)$$

$$\Delta D_2 = \frac{1}{2}T(1 + 1/N)$$

whenever the simulation model uses a zero-order, first-order, or triangular hold. When the simulation model uses some other data hold or combination of data holds, the relative delay may also be computed, by first determining the equivalent data hold resultant from $G(z)_N$.

Returning to the ESM, we illustrate in Fig. 4 the discrete input-output relationship $W(z)$, which contains all of the elements necessary to approximate the time delay in a multirate simulation model. It accurately predicts all of the ΔD_i because of the factor $z^{-1/N}L(z)$, which by itself produces a time delay that is independent of the data hold i:

$$T_W = \frac{1}{2}T(1 + 1/N)$$

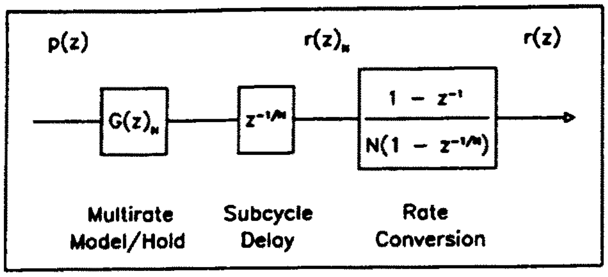

Fig. 4. Multirate portion of ESM.

Considering only the model with its associated data hold $G_i(z)_N$ at the multirate level of computation, it produces (at low frequencies) an advance of $T/2N$ for the zero-order formulation ($i = 0$), an advance of T/N for the first-order formulation ($i = 1$), and no change at all for the triangular formulation ($i = 2$). Since phases add, we see that $W(z)$ predicts the delays

$$\Delta D_0 = T_W - T/2N = \frac{1}{2}T$$

$$\Delta D_1 = T_W - T/N = \frac{1}{2}T(1 - 1/N)$$

$$\Delta D_2 = T_W - 0 = \frac{1}{2}T(1 + 1/N)$$

which are exactly those observed in the time-series analysis.

Similarly, $W(z)$ can be compared to $F(s)$, the continuum model. If its magnitude and phase closely approximate $F(s)$ within the I/O passband, the discrete model exhibits negligible I/O delay. However, to create consistent discrete code, the design objective at the subcycle level is to create a discrete realization such that

$$z^{-1/N}G_i(z)_N = z^{-1/N}\mathcal{Z}\{H_i(s)F(s)\}_N \simeq F(s)$$

If this is accomplished, a well-coded model ensues, because multirate variables are properly advanced for use in succeeding cycles. Unfortunately, this is inconsistent with the elimination of I/O time delays, which is only achievable for single-pass simulations ($N = 1$) because $L(z)$ introduces a time delay for multirate models.

Well-coded models thus display an advance of T/N at the multirate level, as in the cited first-order data-hold case. Unfortunately, they also manifest an I/O time delay entirely caused by the multirate phenomenon. This delay is a consequence of the rate conversion factor $L(z)$ only, because the computer delay per subcycle ($z^{-1/N}$) is canceled by the time advance in $G(z)_N$ for each subcycle. In this case $L(z)$ produces the phase shift

$$\phi = -\frac{1}{2}\omega T(1 - 1/N)$$

or time delay ΔD_1. Hence, for well-coded models, a time delay penalty is invariably associated with multilooping. The delay converges to the value (1/2)T for large N. Since this time delay is constant, it is more properly referred to as a transport delay. "Transport delay" is defined here as the time delay of a discrete realization, obtained from phase measurements relative to the desired continuum system. Well-coded models demonstrate an invariant transport delay over a reasonable range of frequencies.

As has been shown, models not in this class are degraded by both $L(z)$ and the subcycle delay $z^{-1/N}$. For example, formulations that ignore the required advance in $G(z)_N$ show decreasing time delay for increasing N (with the same limit). Hence, if $G(z)_N$ has only the phase characteristics of $F(s)$ and does not include the cyclic advance, the time delay obtained from $z^{-1/N}L(z)$ is ΔD_2.

Models displaying an effective triangular data hold have this delay if an advance, which is needed for the proper synchronization of multirate variables, is not present somewhere in the computations. The Ames simulation system, for instance, has such an advance in its kinematic model.

Models with an effective zero-order data hold accrue a delay of (1/2)T, independent of N. Note that for large N this is the general limit.

If a computationally unstable simulation model can be stabilized by use of alternate algorithms, thereby avoiding aliasing and the multirate delay penalty, the resultant model will probably be equivalent to, or even better than, that obtained using multiple loops, unless "properly designed multirate digital compensators" are implemented (Ref. 3, p. 353). Hence, the delay caused by these more stable algorithms may sometimes be tolerable in simulating networks of transfer functions.

If multirate processes are not used ($N = 1$), the discrete input-output relationship reduces to

$$\frac{r(z)}{p(z)} = z^{-1}\mathcal{Z}\{H(s)_i F(s)\} = z^{-1}G(z)$$

which again shows that the model should be predictive to compensate for the one cycle of delay. It also shows that delays or advances should be represented explicitly in the ESM; they should not be implied within transfer functions. This means that $G(z)$ should represent a concurrent transfer function, such that its output representation explicitly shows its advancing functionality, if any, with respect to the input time.

Concurrent Transfer Functions

Concurrent transfer functions relate the output to the time point of the input; any time shift is shown explicitly in the output representation. An example is given here to avoid the confusion that sometimes arises in reducing difference equations to z-transform notation.

Confusion can be avoided in convoluting difference equations to z-transform notation by taking the most recent input as having the subscript k and letting the output receive its appropriately shifted index. Then variables with the subscript k are assigned the power $z^0 = 1$ in the z-transform representation. For example, the Euler integration algorithm can be represented by

$$v_{k+1} = v_k + Tw_k$$

where the temporal advance is obvious from the subscripts and can be shown explicitly in the z-transform output representation,

$$\frac{zv(z)}{w(z)} = \frac{T}{1 - z^{-1}}$$

In this preferred form the error relative to perfect integration $(1/s)$ is given by the ratio of performance to desired performance,

$$\frac{1}{1/s}\left[\frac{zv(z)}{w(z)}\right] = \frac{j\omega T}{1 - z^{-1}}$$
$$= \frac{\omega T[\sin\omega T + j(1 - \cos\omega T)]}{2(1 - \cos\omega T)}$$

where the phase error is $\phi_\epsilon = (1/2)\omega T$. Since time delay has been defined as the negative of the phase error divided by the frequency, this expression produces a time delay of $-(1/2)T$, or rather, a time advance of $(1/2)T$ with respect to the input point. When the computer delay T is explicitly included, the output at the end of the cycle shows a net delay of $(1/2)T$ with respect to the expected output at the end of the cycle.

The representation of a simulation model using concurrent z-transforms is an important feature of the ESM, in which delays due to procedural code (computer cycles) are shown explicitly. An illustration is provided in the next section, of a model implemented using the Ames techniques.

A Simulation

Consider the second-order transfer function

$$F(s) = \frac{K}{s^2 + Ls + M}$$

where $K = 355.3$, $L = 26.39$, and $M = 355.3$. This system has an undamped natural frequency of 6π (3 Hz) and a damping ratio of 0.7. Because of the constant parameters and the lack of nonlinear elements in this system, the transfer function could be implemented using discrete transfer function techniques. However, for more generality, the parameters are not here considered as constants, and the addition of nonlinear elements (in simulation) is not precluded. A model containing a nonlinearity usually requires that the integration processes be separated.

The second-order system may be illustrated in a discrete realization by explicitly including the computer delays caused by feedback paths and the two integrations $I_A(z)$ and $I_T(z)$, as shown in Fig. 5. The multirate z-transform $G(z)_N$ is thus

$$\frac{z^{1/N}u(z)}{p(z)} = \frac{KI_A(z)_N I_T(z)_N}{1 + z^{-1/N}I_A(z)_N[L + MI_T(z)_N]}$$

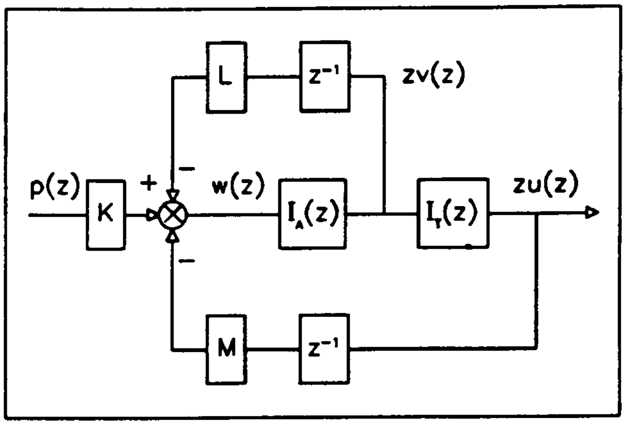

Fig. 5. Second-order model

In this form the predictive responsibility of the integration $I_A(z)$ is well illustrated. In the flight simulation facility at Ames this particular integration is handled by the Adams-Bashforth algorithm, which may be derived from $H_1(s)/s$, and it may be represented by the difference equation

$$v_{k+1} = v_k + \frac{1}{2}T(3w_k - w_{k-1})$$

At the multirate level this has the z-transform (again avoiding confusion)

$$I_A(z)_N = \frac{z^{1/N}v(z)}{w(z)} = \frac{(1/2)\tau(3 - z^{-1/N})}{1 - z^{-1/N}}$$

which manifests considerable phase lead, as required. The second integration algorithm at Ames is derived from the triangular data hold, and it is usually called the trapezoidal algorithm. Since this algorithm is not a predictor, it is naturally in concurrent form:

$$u_k = u_{k-1} + \frac{1}{2}T(v_k + v_{k-1})$$

The z-transform of the trapezoidal algorithm $I_T(z)$ at the multirate level is given by

$$I_T(z)_N = \frac{u(z)}{v(z)} = \frac{(1/2)\tau(1 + z^{-1/N})}{1 - z^{-1/N}}$$

which does not produce phase error with respect to perfect integration. The combination of these two algorithms has two important features pertinent to the discrete realization of vehicle kinematics in the Ames simulation environment. The acceleration-to-velocity integration process absorbs the burden of advancing the phase, approximating a temporal advance of one computer cycle. The velocity-to-position integration algorithm then maintains the correct phase relationship (90° shift), ensuring that consistent velocity and position elements are available. Concomitantly, this scheme relieves programmers of the burden of time-shifting the outputs of various subsystem modules; hence, the triangular hold may be widely used, as recommended.

From the preceding discussion, we see that the total computer delay T is absorbed at the subcycle level by the term $z^{-1/N}$ in the ESM. This is shown in Fig. 3, where the computer delay is given per subcycle. $G(z)_N$ must then manifest the phase lead that is representative of one subcycle, $\tau = T/N$, in addition to the deliberate model phase characteristics of $F(s)$.

Example

As an example of the accuracy of the ESM, we consider the case in which the entire simulation model consists of just the Adams' integration algorithm, $G(z) = I_A(z)$, and we create the error with respect to perfect integration. Consulting Fig. 4, we see that we must include both the subcycle time delay and the rate conversion factor $L(z)$ in a multirate model. Using mixed notation, this produces the relative error

$$E(z) = W(z)/(1/s) = j\omega G(z)_N z^{-1/N} L(z)$$

$$= \frac{j\omega T(3 - z^{-1/N})z^{-1/N}(1 - z^{-1})}{2N^2(1 - z^{-1/N})^2}$$

from which the phase error is calculated:

$$\phi_\epsilon = \tan^{-1}\{\sin(\omega T/N)/[3 - \cos(\omega T/N)]\} - \frac{1}{2}\omega T$$

For determining transport delay, only the low-frequency region is of interest; this is easily extracted from the preceding equation by use of the small-angle ($\omega T/N$) assumption. This produces ΔD_1. Since this is the same as the delay previously produced from examining just the rate conversion function $L(z)$, we here have a well-coded model (manifesting the multirate delay for $N \neq 1$).

The exact phase error can also be determined in this case. This requires an alternate derivation, because for comparison purposes we must avoid the ESM. Consider the sum of a geometric progression,

$$\sum_{n=0}^{N-1} z^{-n/N} = \frac{1 - z^{-1}}{1 - z^{-1/N}}$$

Substitute this into the expression for $I_A(z)_N$. The algorithm then takes the form

$$I_A(z)_N = \frac{1}{2}\tau\left[1 + 2(1 - z^{-1})^{-1}\sum_{n=0}^{N-1} z^{-n/N}\right]$$

For constant inputs for each of N consecutive subcycles, the summation in the above equation collapses to the sum of N constants:

$$\sum_{n=0}^{N-1} z^{-n/N} \Rightarrow N \qquad \text{(with constant inputs)}$$

Use of this substitution eliminates the subcycles from the expression for $I_A(z)_N$, as well as the requirement for rate conversion. That is, using lower brackets ("floor" operator) to denote the least integer operation, the sequence

$$v_{k/N} = v_{(k-1)/N} + \frac{1}{2}\tau\{3p_{\lfloor k/N \rfloor} - p_{\lfloor (k-1)/N \rfloor}\}$$

and the sequence

$$v_K = v_{K-1} + \frac{1}{2}\tau\{(1 + 2N)p_K - p_{K-1}\}$$

deliver the same answers for each $k/N = K$. The entire simulation model, including the total computer delay z^{-1}, then becomes

$$W'(z) = I_A(z)_N z^{-1} = \frac{(1/2)\tau(1 + 2N - z^{-1})}{1 - z^{-1}} z^{-1}$$

This is an exact expression for the response of the discrete system. Its error with respect to perfect integration is given by the relative error

$$E'(z) = W'(z)/(1/s) = j\omega W'(z)$$

which produces a phase error of

$$\phi'_\epsilon = -\tan^{-1}\left[\frac{\tan[(1/2)\omega T][N - \cos(\omega T)]}{1 + N - \cos(\omega T)}\right]$$

Extracting just the small-angle (ωT) performance from this expression also gives the delay ΔD_1.

By comparing z-transforms we can get a feel for the accuracy of the approximations. Forming either $E(z)/E'(z)$ or $W(z)/W'(z)$ and extracting the relative time error gives the results shown in Fig. 6 (the error is zero for $N = 1$).

Figure 6 demonstrates that for frequencies much less than the I/O Nyquist limit, the differences are negligible. Since the "primed" expressions are an exact model of the system, the time delay computed from the ESM in this case displays negligible error.

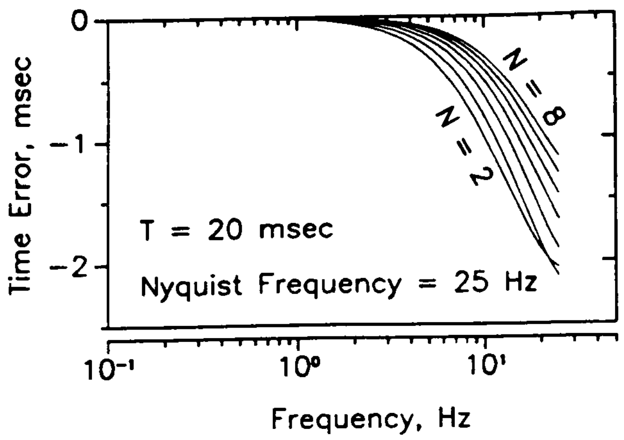

Fig. 6. Adams' ESM error.

The ESM should accurately predict the design phase characteristics for channels represented in z-transform notation. It has been our experience that if this is not the case, then either the model is not well coded, or the cycle time is not compatible with the frequency content of the system.

For more complicated simulation models the analysis of error using the ESM is best handled by time-series analysis, because the algebraic equations become formidable.

Additional Components

The consequences of assumptions (such as linearity) in the ESM are well beyond the scope of this paper, and the reader is invited to explore the limitations of z-transforms in the cited references. However, the benefits of the ESM are indeed great, and for the simulation of flight vehicles the approximations are of little consequence. This is because the spectra of aircraft responses are not competitive with typical I/O Nyquist frequencies in simulation.

The most important feature of the ESM is its multiplicative form for individual components,

$$W(z) = \Pi W_i(z)$$

which means that the individual phases add, as do the effective time delays,

$$D = \Sigma d_i = -\Sigma \phi_i/\omega$$

Each component of the simulation system may thus be considered independently.

We now turn our attention to components other than the simulation model itself and to a demonstration of their inclusion into the multiplicative ESM. An overview of the components included in this model is given by the architecture of Fig. 7, where the time delays ρ_i are caused by the components $W_i(z)$.

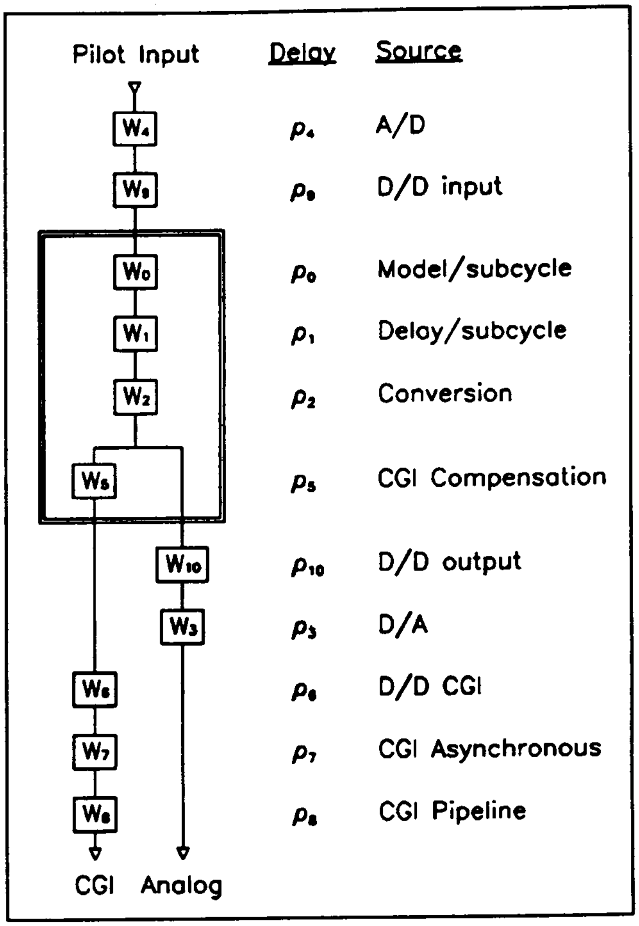

Fig. 7. Equivalent system model.

The following delay sources, listed here in the notation of Fig. 7, have already been discussed: the simulation model per subcycle, $W_0(z) = G(z)_N$, which, when divided by the baseline function $F(s)$, is assumed to have a relatively constant time delay ρ_0 for low frequencies; the transport delay function per subcycle, $W_1(z) = z^{-1/N}$, which produces a time delay $\rho_1 = T/N$ per subcycle; and the conversion function, $W_2(z) = L(z)$, which produces the multirate delay penalty $\rho_2 = (1/2)T(1 - 1/N)$.

Analog Output

The digital-to-analog signal (D/A) is given to a good approximation by

$$r_A(z) = z^{-(1/2)} r(z)$$

This is demonstrated as follows. The Newton-Gregory extrapolation formula for creating representations of data holds is developed in Ref. 2, pp. 31-34. By including just the first term of the formula, the zero-order data hold discussed in the section "The Discrete Multirate Model" is produced, as developed on pp. 34-36 of Ref. 2 and pp. 42-46 of Ref. 3. This may also be written

$$H_0(j\omega) = Th_0(\omega T)e^{(1/2)j\omega T}$$

where the gain factor,

$$h_0(\omega T) = \frac{\sin[(1/2)\omega T]}{(1/2)\omega T}$$

remains close to unity for small ωT products, and is often ignored in sampled-data systems. As developed in Ref. 2, pp. 123-124, when this element is combined with a plant with the transfer function $K(j\omega)$, the overall loop pulse transfer function is a complex expression approximated by

$$HK^*(j\omega) \simeq h_0(\omega T)e^{-(1/2)j\omega T}K(j\omega)$$

when conditions are such that $K(j\omega)$ is low pass relative to the Nyquist criterion. Notice that the factor T disappears. Furthermore, "for low frequencies, $[\sin(1/2)\omega T]/[(1/2)\omega T]$ is approximately unity, so that finally

$$HK^*(j\omega) \simeq e^{-(1/2)j\omega T}K(j\omega)$$

In words, the loop transfer function is approximately the equivalent of replacing the sample and hold operation by a pure time delay of half a sampling period" (Ref. 2, p. 124). Hence, the D/A conversion is

$$W_3(z) = z^{-(1/2)}$$

resulting in the time delay $\rho_3 = (1/2)T$.

Conversion Delay

An analog-to-digital delay function is included in Fig. 7 as $W_4(z)$, producing the time delay ρ_4. Although A/D dynamics produce a delay too small to be of concern here, low-pass prefilters are sometimes used to suppress electrical noise in a facility. Although this noise is generally of high frequency, it aliases into the Nyquist bandwidth during the sampling process. Since low-pass filters usually produce time delays in the low-frequency bandwidth, their phase characteristics must be measured and included in the ESM as the time delay ρ_4.

At Ames Research Center the techniques of common mode rejection have succeeded in reducing laboratory noise. For example, the simulation complex that drives the Vertical Motion Simulator (VMS) does not require A/D prefilters. Hence, in a clean environment such as this, $\rho_4 = 0$.

The values of ρ_6, ρ_9, and ρ_{10} are all digital-to-digital (D/D) delays caused by discrete processors. Each of these components contribute approximately 2 msec of delay in the Ames simulation facility.

CGI Delay

A CGI system usually has projection logic within its first pipeline processor, which accounts for asynchronous data arrival from the simulation computer. The data are usually received via an interrupt into the first processor while the processor continues to work on previously acquired data. The difference between the time of use and time of arrival is used for extrapolation and effectively cancels the asynchronous delay. This delay, given by ρ_7, is therefore approximately zero. If the projection logic were disabled, the asynchronous delay would be approximately one-half the simulation computer's cycle time, $T/2$.

The CGI pipeline delay function, however, is significant. It is given by

$$W_8(z) = z^{-\rho_8/T}$$

The CGI pipeline delay is discussed in Ref. 8. It is generally on the order of 100 msec.

CGI Compensation

The CGI compensation scheme used at Ames is discussed in Ref. 8. It compensates for the pipeline delay ρ_8 by producing a phase lead, equivalent for small frequencies to a time lead ρ_5 (negative valued). From the reference, we see that the compensation algorithm uses the velocity history at the I/O frame rate. The algorithm may be expressed

$$\frac{u_c(z)}{v(z)} = \frac{u(z)}{v(z)} + \sum_{i=1}^{N} b_i z^{1-i}$$

where N is an odd integer, reported in the literature for the case of $N = 3$ (Ref. 8). The b_i coefficients are functions of the cycle time T, the projection interval P, the pipeline delay ρ_8, and a tuned frequency ω_f, nominally set to about 3 Hz (see Ref. 8, p. 7). Using these parameters, CGI pipeline delay essentially vanishes in the pilot's bandwidth, as discussed in the reference, i.e., $\rho_5 + \rho_8 \approx 0$.

Consulting Fig. 7, we can summarize the ESM as follows: (1) The model W_0 contains the desired phase relationships plus one subcycle of time advance. (2) The delay-per-cycle penalty ρ_1 is one subcycle of delay. If the product $W_0 W_1$ constitutes a well-coded model, the delay caused by the discrete realization process vanishes. (3) The D/A penalty ρ_3 is one-half the mainframe cycle time. (4) The A/D penalty ρ_4 is negligible when a prefilter is not required or a good prefilter is used. (5) The CGI compensation algorithm W_5 cancels the pipeline delay ρ_8 for frequencies in the pilot's bandwidth. (6) Digital-to-digital delays are approximately 4 msec; these occur as $\rho_9 + \rho_6$ for the A/D through CGI path, and as $\rho_9 + \rho_{10}$ for the A/D through D/A path. (7) The asynchronous penalty ρ_7 vanishes because of projection logic. (8) The multirate penalty ρ_2 vanishes if multiple loops are not used.

In the Ames flight simulation facility, when a multirate model is not implemented, delay from the pilot to the CGI display is approximately 4 msec and delay from the pilot to the analog outputs of simulations is approximately $4 + T/2$ msec, or about 15 msec. These numbers are considerably less than the 150-msec range of values easily obtained from component-level experiments.

Conclusions

An equivalent system model may be created as the product of individual simulation components expressed as z-transforms,

$$W(z) = \Pi W_i(z)$$

Because of this relationship, the individual phases and component time delays are additive. This permits the independent investigation of various delay sources in flight simulation.

The time-delay penalty for well-coded multirate models (N loops per I/O cycle) is

$$D = \frac{1}{2}T(1 - 1/N)$$

Although procedures for handling multiple loops have been established, the use of a multirate model is not encouraged because of the I/O time-delay penalty and because of problems with aliasing. If alternate solutions are not available for attaining computational stability, then a multirate model should be accompanied by filters that attenuate spectra at the expanded rate.

Simulation input-output relationships that can be expressed in z-transform notation can be included in the equivalent systems model using mathematical relationships described in this paper. Alternately, time-domain software is capable of measuring delays, which may then be used in the equivalent systems model for quality control. A determination may then be made as to whether the model is well coded.

Computer algorithms, notably the predictive integration algorithm and the CGI compensation algorithm, are subsets of the equivalent systems model, as well as subsets of the flight simulation system at Ames. They dramatically reduce total system delays during flight simulation from the delays observed during component-level experiments.

References

[1] Paulk, Clyde Jr., Astill, David L. and Donley, Shawn T., Simulation and Evaluation of the SH-2F Helicopter in a Shipboard Environment Using the Interchangeable Cab System, NASA TM-84387, August 1983.

[2] Ragazzini, John R. and Franklin, Gene F., Sampled-Data Control Systems, McGraw-Hill Book Co., New York, 1958.

[3] Kuo, Benjamin C., Analysis and Synthesis of Sampled-Data Control Systems, Prentice-Hall, Englewood Cliffs, NJ, 1963.

[4] McFarland, R. E., The N/Rev Phenomenon in Simulating a Blade-Element Rotor System, NASA TM-84344, March 1983.

[5] McFarland, R. E., Quiet Mode for Nonlinear Rotor Models, NASA TM-102239, April 1990.

[6] Crochiere, Ronald E., and Rabiner, Lawrence R., Interpolation and Decimation of Digital Signals – A Tutorial Review, Proc. of the IEEE, Vol. 69, No. 3, March 1981.

[7] McFarland, R. E., and Rochkind, A. B., On Optimizing Computations for Transition Matrices, IEEE Trans. Auto. Control, Vol. AC-23, No. 3, June 1978.

[8] McFarland, R. E., Transport Delay Compensation for Computer-Generated Imagery Systems, NASA TM-100084, January 1988.